AMENDMENTS TO THE SPECIFICATION

Page 6:

Please substitute the following paragraph for the paragraph beginning at line 4:

A1

Another object of the invention is to provide a data processor and a data processing system which are minimized in circuit dimensions and yet capable of high accuracy, high speed floating-point number operations ~~highly accurately and at high speed~~.

Page 12:

Please substitute the following paragraph for the paragraph beginning at line 1:

A2

Next will be described in detail the data processor DP illustrated in Fig. 1. In this embodiment of the data processor ~~in this embodiment~~, the SIMD type floating-point unit FPU has no memory addressing capability ~~so as~~ in order to ~~save~~ reduce the area it occupies. In other words, the central processing unit CPU, instead of the SIMD type floating-point unit, has a memory addressing function. For this reason, the central processing unit not only fetches data from the memory for the SIMD type floating-point unit but also fetches all the instructions including the

floating-point instruction for the SIMD type floating-point unit from the memory. The instructions fetched by the central processing unit are taken in by both the central processing unit and the SIMD type floating-point unit via the instruction bus IB and decoded. The central processing unit, if the decoded instruction is a CPU instruction, executes integer processing in accordance with the decoded instruction or, if the decoded instruction is a floating-point instruction, executes addressing processing or the like on behalf of the SIMD type floating-point unit. ~~The~~ If the decoded instruction is a CPU instruction, the SIMD type floating-point unit~~, if the decoded instruction is a CPU instruction,~~ ignores the instruction~~ or,~~, or if the decoded instruction is a floating-point instruction, the SIMD type floating-point unit carries out the floating-point calculation in accordance with the decoded instruction. Here, if the decoded instruction is either a load or store instruction, the central processing unit outputs the data address to the data cache to request loading or storage of data. In response to that request, the data cache DCA loads the data of the inputted data address onto the data bus DB or stores them from the data bus as the case may be. Incidentally, even where the data are to be stored, the external address of a usual copy-back

A2 cont.

cache is for reading, and the storage is done onto part of a cache line cached by the reading, though if the cache line is replaced with an effective and updated one at the time of caching, the cache line is externally copied back. If the register for which the load or store instruction is destined is a register of the SIMD type floating-point unit, the SIMD type floating-point unit, if the instruction is to load, writes the value on the data bus into the register of the SIMD type floating-point unit or if the instruction is to store, outputs the value of the register in the SIMD type floating-point unit onto the data bus. Incidentally, the loading and storage by the SIMD type floating-point unit will be described with reference to Fig. 3.

A2 Cont'd.

Page 16:

Please substitute the following paragraph for the paragraph beginning at line 5:

There are two conceivable methods to define the register: one way of defining it is by dividing a register file that is 32 bits wide into four banks by the register number, and the other way of defining it is by dividing a 128 bit wide register ~~of 128 bits in width~~ bit wise into four ~~in the bit direction~~banks. The advantage of the

A3

former is that a length-4 vector can be defined in a width of 32 bits, while that of the latter is that a length-4 vector can be defined with a single register. However, although the latter facilitates ~~the processing of~~ reading out of or writing into the register and forwarding, because data, whether vector or scalar, can be accommodated by a single register, the efficiency of use and the extensibility of the register deteriorates. For instance, where an inner product instruction is defined for a register of 128 bits in width, the 96 most significant bits will be wasted because the register to store the operation output is also 128 bits in width. Or if a 128 bit width is used for defining the inner product instruction, it will become difficult to compose the inner product instruction in SIMD. For instance, four in parallel would require a 512 bit width. On the other hand, the former arrangement, ~~as it makes possible definition of~~ by defining a length-4 vector inner product instruction in a 32 bit width, involves no waste of the output register, and the instruction can be easily composed in SIMD. These considerations reveal that the level of parallelism can be enhanced by defining a length-4 vector by dividing the register file into four banks by the register number and defining an inner product instruction relative to that

A3 cont'd

length-4 vector, and composing the register file and the inner product instruction in SIMD, and it is made possible to enhance the level of parallelism per instruction efficiently. The former method is adopted for this embodiment of the invention.

A3 cont'd.

Page 18:

Please substitute the following paragraph for the paragraph beginning at line 18:

A4

On each bank, one out of the sixteen 32 bit registers designated by the four most significant bits 41 of read register numbers RNW, ~~RNX , RNY~~ RNX, RNY and RNZ. Since this embodiment has four banks, it is possible to read and output data out of a total of four registers. The transfer 4 bank read port W, when the transfer width is less than 128 bits, directly supplies the output of each bank because the transfer block TBLK performs alignment then. The 4 bank read ports X and Y, ~~as does the~~ and the transfer 4 bank read port~~,~~ directly supply the output of each bank at the time of 128 bit outputting for vector instruction. At the time of 32 bit outputting for usual operation, the two least significant bits 42 of the register numbers RNW, RNX, RNY and RNZ are used, the bank to be read out of is selected with selectors 432 and 433, and outputting to X[0]

A4 cont'd

and Y[0] is accomplished. Since the conventional read port Z is a usual 32 bit port, it always uses the two least significant bits 42 of the register numbers RNW, RNX, RNY and RNZ to select the bank to be read out of with a selector 431, and ~~outputs~~ output data. As a result, it is ~~made~~ possible to mount the values of any three out of the 64 registers on X[0], Y[0] and Z.

Page 19:

Please substitute the following paragraph for the paragraph beginning at line 14:

A5

~~To add~~Additionally, since the ports of the register file RGST in this embodiment are divided into the transfer 4 bank write port U, the transfer 4 bank read port W, the operation write port V, the 4 bank read ports X and Y, and the conventional read port Z, it is also possible to apply a superscalar or a very long instruction word (VLIW) architecture.

Page 20:

Please substitute the following paragraph for the paragraph beginning at line 6:

A6

~~Of~~ For each of the X or Y floating-point numbers that ~~is~~ are entered, the sign is ~~inputted~~ provided as input to a

A6 cont'd.

sign processor SPP~~;~~, the exponent~~,~~ is provided as input to an exponent processor EPP~~;~~, and the mantissa~~,~~ is provided as input to the multipliers (MLP0, MLP1, MLP2 and MLP3).

Page 23:

Please substitute the following paragraph for the paragraph beginning at line 8:

A7

Fig. 7 illustrates the configuration of the exponent difference generator EDG. It calculates with ten 4 input adders FADR the exponent difference between every pair of terms that can be selected out of five terms. For instance, the exponent difference between X[0] × Y[0] and X[1] × Y[1] is EX[0] + EY[0] - EX[1] - EY[1]. Every exponent is biased with "127" based on a standard. However, as four biases cancel each other in the above-cited formula, the bias of any exponent difference calculated by this formula is "0". On the other hand, the exponent difference between X[3] × Y[3] and Z is EX[3] + EY[3] - EZ - 127. The subtraction of ~~128~~ 127 is to cancel the bias.

Page 26:

Please substitute the following paragraph for the paragraph beginning at line 23:

With reference to Fig. 12, the 9 input adder ADDR shown in Fig. 5 will now be described in detail. In this embodiment, in spite of the 9 input addition arrangement, the increase in the number of digits is at most three bits because the carry save form increases five inputs to the nine inputs. First, each of the nine inputs supplied by the aligners ALN0, ALN1, ALN2, ALN3 and ALNZ are extended by three bit signs by a sign extender SE. The output of the sign extender is inputted to a 9 input carry save adder array CSA. Then, carry-in is accomplished according to the number of terms negated. As two bits are carried in to invert one pair of products in the carry save form, a maximum of eight bits should be carried in for four products. Since negation is controlled with Inv[0], Inv[1], Inv[2] and Inv[3], a 2 bit carry is matched with each of these four signals as illustrated, 6 bit and 2 bit ~~carries-in~~ carry-ins are performed on the 9 input carry save adder array CSA and the carry propagate adder CPA, respectively. The carry propagate adder generates a pre-normalization mantissa Macm from the aforementioned 2 bit

carries, and a carry output COUT and a sum output SOUT from the 9 input carry save adder array.

A8 cont'd

Page 27:

Please substitute the following paragraph for the paragraph beginning at line 19:

A9

Fig. 13 illustrates in detail the 9 input carry save adder array CSA shown in Fig. 12. From the left side in Fig. 13, the output of the 3 bit sign extender is inputted. Therein the first stage consists of 3 input carry save adders 730, 731 and 732, the second stage, also of 3 input carry save adders 733 and 734, and the third stage of a 4 input carry save adder 735. This configuration serves to reduce the number of terms from 9 to 6, then to 4, ~~and~~ then to 2, and finally to provide the carry output COUT and the sum output SOUT. In this embodiment, carry-in is possible up to six bits of CI0, CI1, CI2, CI3, CI4 and CI5. To add, the 3 input carry save adder is composed by arranging as many 1 bit 3 input carry save adders, such as the one shown in Fig. 14 for instance, as matching the bit width. Further, the 4 input carry save adder is composed of 1 bit 4 input carry save adders, such as the one shown in Fig. 15.

Page 29:

Please substitute the following paragraph for the paragraph beginning at line 1:

~~To the~~The aforementioned bus BUS ~~are~~ is connected to an SRAM to be used as the working area of the data processor or an area for temporary storage of data, and a ROM on which the operating system (OS) of the data processor and the like are stored. A DRAM is also connected to the bus via a control circuit DMC. The control circuit DMC, which is to perform address multiplex control and refresh control over the DRAM, may have a configuration in which its functions are distributed within the DRAM or within the data processor. Further, a peripheral device control unit PDC and a display controller DC are connected to the bus. To the peripheral device control unit are connected an external storage device ESD, such as an optical disk, a keyboard KBD and the like. A display unit DP is connected to the display controller.

A10

Please substitute the following paragraph for the paragraph beginning at line 15:

The above-described data processor, as it is provided with instructions for executing floating-point operations and has registers for floating-point operations to transfer

A11

instructions, can execute ~~at~~ high speed floating-point number operations which are frequently used in three dimensional graphic processing. Therefore the data processing system embodying the invention, as described above, ~~to~~ can be used as a game machine, a portable information terminal or the like, which are items of multimedia equipment, makes possible three dimensional graphic processing with high precision and at high speed while reducing the overall cost.

A11 cont'd

Page 30:

Please substitute the following paragraph for the paragraph beginning at line 3:

A12

It is further possible, in the data processing system of Fig. 16, to add a rendering coprocessor to the bus BUS. Three dimensional graphic processing consists of geometric processing and rendering. Geometry, which uses many inner product operations and vector converting operations, is ~~let be~~ processed by the data processor DP embodying the invention as described above, while rendering is left to the rendering coprocessor. This disposition makes it possible to provide a data processing system capable of accomplishing three dimensional graphic processing faster than a data processing system which causes rendering to be

processed by a central processing unit within its data processor.

A12 cont'd

Page 33:

Please substitute the following paragraph for the paragraph beginning at line 12:

A13

~~By composing~~ Composing the instruction to multiply four length-4 vectors in parallel in SIMD ~~of four in parallel~~ according to the invention, ~~it is made possible to execute~~ allows execution of 28 operations ~~at~~ in a single instruction. Furthermore, ~~by~~ defining the processing to add the inner product of the length-4 vectors and scalars, ~~it is made possible~~ allows the processor to handle hypercomplex vector data ~~of over~~ greater than length-4, and to execute 32 operations ~~at~~ in a single instruction. Therefore, it is made possible to provide a data processor capable of processing ~~the operation of~~ floating-point ~~numbers~~ number operations at high speed and, in addition, to provide a data processing system capable of multimedia processing, in particular, high speed three dimensional graphic processing ~~at high speed~~.